So-Lo
Team 12

Andy Weng (CSE)  ★
Suzet Nkwaya (EE)
Ming Shuai Chen (CSE)
Dan-Michael Tiamzon (EE)

UMass Amherst
Department of Electrical and Computer Engineering
Advisor: Baird Soules
Current Problems with Recording Meetings

- Recording group meetings
  - Long meetings are full of information
  - Often meetings are recorded for audio only
  - Camera is usually in a fixed position for A/V recording
So-Lo

- Real time sound locator
- Rotating stand that points almost to the location of sound
- Effective for small to medium sized rooms
- Utilizing the right microphone sensitivity
Previous Block Diagram

- Microphones
  - Preamplifier
  - Filters
  - Amplifier
  - ADC

- Software
  - Motor Control
  - Angle Calculator
  - Save Motor Locations

- Computer
  - Camera (Mounted)
  - Motor

- Save Recording

Legend
Suzet Nkwaya
Andy Weng
Ming Chen
Dan Tiamzon
Dan and Suzet
Changes in Implementation

● Removal of ADC
  ○ ADC is too slow to sample and communicate with Pi

● New Implementation
  ○ Much Faster
    ○ The amplified and filtered signal from the microphone is directed to a comparator
    ○ Compare the input sound to noise level
    ○ Produce high if sound is higher than noise level
    ○ Send this signal directly into the Raspberry Pi
MDR Deliverables (Updated)

- Mainly present the concept of sound location
  - Set-up microphone array to sense sounds. (Suzet)
  - Raspberry Pi and Python Code to determine the order of which microphones receive signals. (Dan)
  - Code which implements TDOA using the order of microphones and estimate sound source location angle. (Ming)
  - Control rotation angle of the motor. (Andy)

- No implementation of video recording and saving.
  - Does not present the concept of sound locating.
System Requirements

- Calculate source of sound based on time differences between microphones located at a known position.

- Microphone Array
  - amplify voice and filter out noise.
    - Amplifier: TL074 CN
    - Filter: Bandpass 100Hz-500Hz
    - Comparator: LM 311N
System Requirements

- Raspberry Pi
  - use microphone array outputs to calculate sound source location

- Motor
  - Receive commands from Raspberry Pi

- Camera
  - Record video and save data on SD
Microphone Array Circuit
Amplifier

[Diagram of an amplifier circuit]
Comparator
Time Difference of Arrival

- TDOA of 3 microphones analyzed on a Logic Analyzer
Interrupt System

Software that detects for interrupts on the Raspberry Pi GPIO channels.

- Software is written in Python.
- Input = Voltage produced by microphone system.
- Output = Order in which the microphones received a signal and the time differences between the 1st & 2nd microphones and the 1st & 3rd microphones.
- Output will be sent to the software which calculates the angle in which the sound originated from.
Interrupt System

- Setup 3 GPIO pins to detect events. Events to be detected is a voltage.
- Connect the 3 GPIO pins to terminals of a 4 Pin Dip Switch.
- 3.3V Pin from Pi connected to other terminals of Dip Switch.
- Flip all switches at the same time closely.
- Code marks which channel receives a voltage first and marks the time that voltage arrived.
- Code keeps track of the order of arrival and solve the time differences.
TDOA (Time Difference of Arrival)

TDOA utilizes the equation of a circle to pinpoint the source. Equation of a circle: \((x - h)^2 + (y - k)^2 = r^2\)
Implementation of TDOA in Python

»Origin at x=1000, y=1000
Implementation of TDOA in Python

»Origin at x=1000, y=1000
»Microphones are set up roughly 12 inches apart.
Implementation of TDOA in Python

» Origin at x=1000, y=1000
» Microphones are set up roughly 12 inches apart.
» Program takes in x pos and y pos of the speaker.
Implementation of TDOA in Python

-Origin at x=1000, y=1000
-Microphones are set up roughly 12 inches apart.
-Program takes in x pos and y pos of the speaker.
-Calculates the time bs and cs, time of second and third order microphone.
Implementation of TDOA in Python

- Origin at x=1000, y=1000
- Microphones are set up roughly 12 inches apart.
- Program takes in x pos and y pos of the speaker.
- Calculates the time bs and cs, time of second and third order microphone.
- Program outputs calculated x and y position.
Implementation of TDOA in Python

-Origin at x=1000, y=1000
-Microphones are set up roughly 12 inches apart.
-Program takes in x pos and y pos of the speaker.
-Calculates the time bs and cs, time of second and third order microphone.
-Program outputs calculated x and y position
-This should match our input
Implementation of TDOA in Python

» Origin at x=1000, y=1000
» Microphones are set up roughly 12 inches apart.
» Program takes in x pos and y pos of the speaker.
» Calculates the time bs and cs, time of second and third order microphone.
» Program outputs calculated x and y position
» This should match our input
» Angle is calculated
L293D Motor Driver

- Raspberry Pi not capable of driving the motor we are using
- Capable of driving two motors (only one is used)
- Allows us to control the direction the motor spins by reversing the current going through the motor
The Encoder

- 5 pins - Vcc, Gnd, A, B, Index
- Only used A channel
- Disc connected to a shaft that is also connected to the motor
- When the disc spins, channel A outputs a series of pulses
Implementation of the Motor
Implementation of the Motor

![Diagram showing rotation angles (0°, 90°, 180°, 270°)]
Implementation of the Motor
Proposed CDR Deliverables

- Demonstration of Complete System Functionality (Team)

- Place the microphone arrays in an equilateral triangle.

- Detect voice from 3 feet away (Suzet)

- Motor will respond and turn to the angle produced by the angle calculator (Andy & Dan)

- Record and store 30 second video on SD card (Ming)
Microphone Demo (backup)